

# Learning Support by Reflection and Knowledge Collaboration in a Team-based Software Engineering Project Course - Position paper -

Atsuo HAZEYAMA

Department of Information Science, Tokyo Gakugei University

*E-mail: hazeyama@u-gakugei.ac.jp*

## Abstract

*Software development is a highly knowledge-intensive and collaborative activity. Problem resolution processes are performed iteratively during software development. We propose a learning model that is based on reflection and knowledge collaboration for problem resolution in a software engineering project course. We also describe an overview of a support system based on this model.*

## 1. Introduction

Software development is a highly knowledge-intensive [8] and collaborative activity [1]. Problem resolution processes are performed iteratively during software development. Individual developers possess only a specific part of the knowledge and expertise required for software development. Therefore, they develop software and simultaneously collect various types of information. Some developers even consult experts [9]. Ye emphasizes the importance of knowledge collaboration in software engineering [9]. For this purpose, he provides the following three types of support facilities for a social platform: finding sample programs, browsing the archives of previous discussions, and posing questions to selected experts.

We tackle studies on a team-based software engineering project course [3]. There it is important for learners to acquire knowledge and skills on software development. I propose to introduce reflection support as well as knowledge collaboration for the domain. Reid defined reflection as being a process of reviewing an experience of practice in order to describe, analyze, evaluate and so inform learning about practice [6]. Sample programs, discussions with respect to past problem solving, and/or answers to questions will be important information for a learner's problem solving. In addition, I believe reflection of his/her problem solving and recording of the process enhance his/her understanding. These processes correspond to internalization and externalization of the SECI model [5]. Furthermore by sharing the information that was described as the result of reflection, it may be able to contribute to problem solving of other learners. We regard problem solving that occurs during software

development as a kind of learning. Therefore I think it is also useful for software engineers to describe their reflection process in professional software development.

Hazzan described significance of introducing reflective perspective [7] and studio concept into software engineering education [4]. That paper described the process, which facilitates reflection. However, it did not clarify how results of reflection are externalized and the supporting environment.

## 2. Proposal

This section proposes a conceptual framework for learning support by reflection and knowledge collaboration in a team-based software engineering project course.

### 2.1 Conceptual model

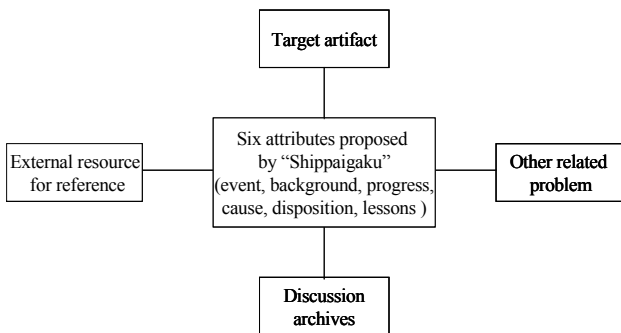
When a person encounters a problem, the following three problem solving patterns are considered:

- (1) Solving the problem by himself/herself
- (2) Solving the problem over searching for related information and referring to it
- (3) Solving the problem by posing questions to others

“To present sample programs” and “to browse past archived discussions” out of the three facilities Ye provided correspond to (2). “To pose questions to selected experts” correspond to (3). This study supports the abovementioned three patterns. Even if which pattern is adopted, I ask the learner for reflection and describing the result after problem resolution. Through the process, I aim at enhancing his/her knowledge. Especially when a learner solved his/her problem by referring to past archived discussions and/or sample programs other created, or by advices from other learners, reflection improves his/her understanding and it enhances the information the learner referred. This process corresponds to combination of the SECI model. This information is valuable when another learner reuses it.

Figure 1 shows information structure for learning support by reflection and knowledge collaboration. It is consisted of five major objects, the problem

concerned and description by reflection for it, target artifacts that included the problem and that the problem has been fixed, external resources that were referred for problem solving, discussion archives with others, and other problems that were related to the concerned problem. We adopted Shippaigaku's attributes for reflection [2]. Shippaigaku is a theory whose goal is to learn from failures. It aims at avoiding similar mistakes and/or accidents to past ones by learning. Shippaigaku defines six attributes; a problem description (event) and its accompanied descriptions (background and progress), the information toward problem resolution (cause and disposition) and lessons learned from the problem solving. In addition to the six attributes of Shippaigaku, I manage the following information and associate them with the problem-solving information: the result of disposition as the target artifact object, external resources, discussion archives, and other related problems.



**Figure 1. Information structure for our learning support environment**

## 2.2 Knowledge sharing support

As the volume of knowledge accumulated in the environment increases, it is difficult to retrieve appropriate information in an efficient manner. Most knowledge management systems utilize voting information and/or access log information for this purpose [10]. In addition, I associate a problem with usecase, which is unit of a function a system has because we suppose similar tasks may have similar problems. We use a usecase name in order to specify functionality.

## 3. Summary

We have proposed a learning model for the problem resolution process in a team-based software engineering project course; this model integrates reflection with knowledge collaboration. We will implement this model and apply it to an actual software engineering project course to validate the proposed framework. We will also evaluate a

trade-off problem between the cost to describe the reflection process and learning effectiveness.

## Acknowledgments

The author would like to thank anonymous reviewers for their comments to improve this paper. This study is supported by the Grant-in Aid for No. (C) 18500701 from The Ministry of Education, Science, Sports and Culture of Japan.

## References

- [1] D. M. German, D. Cubranic, and M-A. D. Storey, "A Framework for Describing and Understanding Mining Tools in Software Development", *Proceedings of the 2nd International Workshop on Mining Software Repositories (MSR 2005)*, pp. 95-99, ACM Press, 2005.
- [2] Y. Hatamura, *Shippaigaku no susume*, Kodansha, 2000 (in Japanese).
- [3] A. Hazeyama, "An Education Class on Design and Implementation of an Information System in a University and Its Evaluation", *Proceedings of the 24th Annual International Computer Software and Applications Conference (COMPSAC2000)*, IEEE CS Press, pp. 21 - 27, October 2000.
- [4] O. Hazzan, "The reflective practitioner perspective in software engineering education", *The Journal of Systems and Software*, Vol. 63, pp. 161-171, 2002.
- [5] I. Nonaka, and H. Takeuchi, *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation*, Oxford University Press, 1995.
- [6] B. Reid, "But we're doing it already" Exploring a response to the concept of reflective practice in order to improve its facilitation. *Nurse Ed Today*, Vol. 13, pp. 305-309, 1993.
- [7] D. A. Schon, *The Reflective Practitioner: How Professionals Think in Action*, Basic Books, 1983.
- [8] Y. Ye, and K. Kishida, 1<sup>st</sup> International Workshop on Supporting Knowledge Collaboration in Software Development, in *Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC2005)*, 2005, <http://13d.cs.colorado.edu/%7Eyunwen/KCSD2005/>.
- [9] Y. Ye, "Socio-Technical Support for Knowledge Collaboration in Software Development Tools", *Proceedings of the Workshop on Integrating Software Engineering and Usability*, pp. 39-51, 2005.
- [10] M. Zacklad, "Communities of action: a cognitive and social approach to the design of CSCW systems", *Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work*, pp. 190 - 197, ACM Press, 2003.